

## Distance-Based Classification Methods

Oya Ekin, Peter L. Hammer, Alexander Kogan & Pawel Winter

To cite this article: Oya Ekin, Peter L. Hammer, Alexander Kogan & Pawel Winter (1999)  
Distance-Based Classification Methods, INFOR: Information Systems and Operational Research,  
37:3, 337-352, DOI: [10.1080/03155986.1999.11732388](https://doi.org/10.1080/03155986.1999.11732388)

To link to this article: <https://doi.org/10.1080/03155986.1999.11732388>



Published online: 25 May 2016.



---

Submit your article to this journal [↗](#)



---

Article views: 61



---

Citing articles: 1 View citing articles [↗](#)

---

# DISTANCE-BASED CLASSIFICATION METHODS<sup>1</sup>

OYA EKIN

*Department of Industrial Engineering, Bilkent University, Bilkent, 06533 Ankara, Turkey.  
karasan@Bilkent.EDU.TR*

PETER L. HAMMER

*RUTCOR, Rutgers University, 640 Bartholomew Road, Piscataway,  
New Jersey, 08854-8003, U.S.A.  
hammer@rutcor.rutgers.edu*

ALEXANDER KOGAN

*Accounting and Information Systems, Faculty of Management, Rutgers University,  
180 University Ave., Newark, New Jersey 07102, U.S.A.  
kogan@rutcor.rutgers.edu*

PAWEL WINTER

*Department of Computer Science, University of Copenhagen,  
DK-2100, Copenhagen O, Denmark.  
pawel@diku.dk*

## ABSTRACT

Given a set of points in a Euclidean space, and a partitioning of this "training set" into two or more subsets ("classes"), we consider the problem of identifying a "reasonable" assignment of another point in the Euclidean space ("query point") to one of these classes. The various classifications proposed in this paper are determined by the distances between the query point and the points in the training set.

We report results of extensive computational experiments comparing the new methods with two well-known distance-based classification methods ( $k$ -nearest neighbors and Parzen windows) on data sets commonly used in the literature. The results show that the performance of both new and old distance-based methods is on par with and often better than that of the other best classification methods known. Moreover, the new classification procedures proposed in this paper are: (i) easy to implement, (ii) extremely fast, and (iii) very robust (i.e. their performance is insignificantly affected by the choice of parameter values).

## RÉSUMÉ

Étant donné un ensemble de points d'un espace Éuclidien, ainsi qu'une partition de cet "ensemble d'apprentissage" en deux ou plusieurs sous-ensembles ("classes"), on se propose d'identifier une affectation "raisonnable" d'un point n'appartenant pas à l'ensemble d'apprentissage à l'une de ces classes. Les différentes classifications considérées dans cet article sont définies en fonction des distances entre le point de classification inconnue et les points de l'ensemble d'apprentissage.

On présente les résultats d'une série d'expériences computationnelles effectuées sur plusieurs ensembles de données fréquemment utilisées dans la littérature. On compare nos méthodes avec deux méthodes à base de distances bien connues – celle des voisins d'ordre  $k$  et celle des fenêtres de Parzen. Nos résultats montrent que les performances de toutes les méthodes à base de distances examinées sont comparables et souvent supérieures aux performances des meilleures autres méthodes de classification. De plus, les nouvelles procédures de classification que nous proposons sont: (i) facilement applicables, (ii) extrêmement rapides et (iii) très stables (i.e. leur performance ne dépendant pas significativement des changements mineurs dans les valeurs des paramètres).

<sup>1</sup>Recd. Jan. 1998; Revd. Feb. 1999

## 1. INTRODUCTION

In this paper we suggest and study several new classification methods. The roots of some of these methods can be traced back to the techniques that were popular during the early days of pattern recognition. Given a set of points in a Euclidean space, and a partitioning of this "training set" into two or more subsets ("classes"), we study here the problem of identifying a "reasonable" assignment of another point in the Euclidean space ("query point") to one of these classes.

The methods described here attempt to (i) find for a query point a "reference group" in each class of the training set, (ii) choose among them a "best" reference group, and then (iii) classify the query point accordingly. The best known technique of this type is the *nearest neighbor* method. The work of Fix and Hodges [16] on nonparametric statistical pattern classification is recognized as having pioneered the nearest neighbor classification techniques. Cover and Hart [10] investigated the nearest neighbor procedure as a tool for pattern recognition and established its asymptotic performance. Since its introduction, this procedure has been extensively used, and numerous variants of it have been suggested. Some of these studies concentrate on algorithmic developments and implementation aspects [15] of the method while others generalize it to modified metrics [17].

Learning to classify objects is a fundamental problem in artificial intelligence. In the literature, variants of the nearest neighbor method and case-based learning algorithms have been extensively researched [1, 9, 11, 10, 12, 13, 17, 37]. Case-based learning algorithms input a sequence of training points and output a "concept description", which can be used to generate predictions of class values for subsequently presented query points. For numeric attribute values, normalized Euclidean distances have been commonly used to compare points. For learning in domains in which some or all of the training examples are symbolic, special variants of nearest neighbors algorithms have been designed [9]. The way of choosing the reference groups and the method of evaluating the proximity of the query point to each of them differentiate one technique from another.

The various classifications proposed in this paper are determined by the distances between the query point and the points in the training set. In this respect they may be considered to be of the same flavor as the case-based reasoning techniques of artificial intelligence and the k-nearest neighbors, although the mechanics differ substantially.

Section 2 starts with a precise formulation of the classification problem and proceeds to describe the two best known distance based classification techniques: the k-nearest neighbors and the Parzen windows methods. Section 3 is devoted to a detailed description of the new distance based classification techniques proposed in this paper: the adjusted averaging method, the adjusted weighting method, the truncated potentials method and the convex containment method. In Section 4 we present brief descriptions of the 9 datasets from the UC Irvine repository of machine learning datasets that were used in this paper to evaluate computationally the performance of the distance based classification techniques. Section 5 describes the experimental method we used in our computations, while Section 6 presents the results we obtained in our experiments, including the sensitivity analysis studies showing how the examined methods are affected by the choice of the parameter value. Finally, in Section 7 we present the conclusions of our study:

- The methods described in this paper are extremely efficient; they require no extensive preprocessing since they do not build domain theories explicitly.
- These methods are robust and very accurate, as shown by the computational experiments.

- There is no major difference indicating very clearly that a particular variant of these methods performs in a significantly and consistently superior way than the others.

## 2. PROBLEM FORMULATION

We consider a multi-class classification problem in which points are described by vectors of attribute values. We assume that we are given  $m$  subsets  $S_1, S_2, \dots, S_m$  ( $m \geq 2$ ) of  $d$ -dimensional vectors. The cardinality of  $S_i$ ,  $1 \leq i \leq m$ , is denoted by  $n_i$ . We assume that  $S_1, S_2, \dots, S_m$  are pairwise disjoint and the total number of points in all subsets is denoted by  $n$ . Finally,  $n^* = \min_{1 \leq i \leq m} \{n_i\}$ . Each vector can be represented as a point in a  $d$ -dimensional space  $E^d$ . If the vectors are binary, they correspond to vertices of a  $d$ -dimensional hypercube  $H^d$ . The  $L_p$  distance metrics are defined by

$$D_p(x, y) = \left( \sum_{i=1}^d |x_i - y_i|^p \right)^{1/p}, \text{ where } x, y \in E^d.$$

Here we shall restrict ourselves to the cases  $p = 1$  or  $2$ .

We would like to be able to classify any point in  $E^d$  as belonging to one of the  $m$  classes. Furthermore, this classification should keep together in some way points that are “intuitively” of the same type.

For  $d$  relatively small ( $d \leq 20$ ), when the query points are assumed to be vertices of  $H^d$ , the partitioning could be explicit, i.e., each of the  $2^d$  vertices could be stored along with an integer specifying the class it belongs to. Deciding whether a query point is “red” or “blue” can be done in  $O(d)$  time.

On the other hand, for larger  $d$ , and for non-discrete domains, implicit partitioning is required, i.e. a more compact description of the partition must be generated. In this case, when query points arrive, additional computation, rather than a trivial lookup, is needed to determine the classification.

The amount of this computation depends on the way in which the classification rule is defined. When the classification rule has an analytic form, the bulk of the computations was already carried out at the time when the analytic form was developed (preprocessing stage).

At the other end of the spectrum is the situation when no preprocessing is needed, and when the classification of a query point is determined directly from the original training set. This paper focuses on this latter kind of methods. Two well-known procedures of this type are briefly described below, while the new approaches are described in Section 3. The computational experiments and their results are discussed in Sections 5 and 6.

### 2.1 *k*-Nearest Neighbor (*k*-NN)

One of the simplest ways to classify a query point  $q$  is based on the class of its nearest neighbor in the training set. A straightforward generalization of this approach is to classify a query point as belonging to the class which is most frequently represented among its  $k$  nearest neighbors. The choice of  $k$  is data dependent. Typically,  $k$  should depend on the size of the training set (e.g.  $k = \sqrt{n}$ ). The  $L_2$  distance is the most commonly used metric, but Fukunaga and Flick [17] have proposed a global quadratic metric that can outperform  $L_2$ . A number of variants of this *k*-nearest neighbors approach have been suggested in the literature (for an extensive survey see [11]).

### 2.2 Parzen Windows (Parzen)

Another standard distance-based classification method, known as the *Parzen windows* method (see [25]) is the following. As a query point  $q$  arrives, a “window” of some

prespecified size, centered at  $q$ , is introduced. In the simplest variant, this window is a  $d$ -dimensional ball centered at  $q$  and having radius  $r$ . The number of points of each class contained within the window is counted, and the query point is classified as belonging to the class with the maximum number of occurrences within the window.

### 3. NEW DISTANCE-BASED CLASSIFIERS

Throughout this section we assume that for a fixed query point  $q$  the points in class  $S_c$ ,  $1 \leq c \leq m$ , are indexed  $p_1^c, p_2^c, \dots, p_{n_c}^c$  so that  $d(q, p_1^c) \leq d(q, p_2^c) \leq \dots \leq d(q, p_{n_c}^c)$ . Unless otherwise stated, we assume  $d$  to be the  $L_1$  distance function.

#### 3.1 Adjusted Averaging Method (AAM)

Let us define

$$f_\alpha^c(i, q) = \frac{\sum_{j=1}^i d(q, p_j^c)}{i - \alpha}$$

for  $i = \alpha + 1, \alpha + 2, \dots, n_c$ , and an appropriately chosen integer  $\alpha$ ,  $0 \leq \alpha < n_c$ . Let

$$F_\alpha^c(q) = \min_{\alpha+1 \leq i \leq n_c} \{f_\alpha^c(i, q)\}.$$

A query point  $q$  is considered to belong to  $S_c$  if  $F_\alpha^c(q) \leq F_{\alpha'}^c(q)$  for all  $c'$ ,  $1 \leq c' \leq m$ . It can be shown (see Appendix) that the sequence

$$f_\alpha^c(\alpha + 1, q), f_\alpha^c(\alpha + 2, q), \dots, f_\alpha^c(n_c, q)$$

is unimodal for any  $\alpha \geq 0$ . The unimodality of the  $f_\alpha^c$ -sequence simplifies considerably the determination of  $F_\alpha^c(q)$  as there is no need to consider the  $f_\alpha^c$ -values once they start increasing.

If  $\alpha = 0$ , this classification rule is the 1-nearest neighbor rule, since  $f_0^c(1, q) = d(q, p_1^c)$ , while  $f_0^c(2, q) = (d(q, p_1^c) + d(q, p_2^c))/2 \geq d(q, p_1^c)$ . Hence,  $F_0^c(q) = f_0^c(1, q)$ .

Our motivation for introducing this classification method is closely related to the Steiner tree problem. Let  $S$  be a set of points in an arbitrary metric space. The *Steiner tree problem* for  $S$  is to find the shortest possible network  $T$  spanning all the points in  $S$ . The edges of  $T$  are not limited to meet at the  $S$ -points only.

Recall that  $S_c$ ,  $1 \leq c \leq m$  denotes the set of points of the class  $c$  in the training set. A reasonable strategy for classifying a query point  $q$  is to consider the ratios of the lengths of the minimal Steiner trees for  $S_c$  and for  $S_c \cup \{q\}$ ,  $1 \leq c \leq m$ . The query point  $q$  is then classified as belonging to the class with the smallest ratio. This strategy is very expensive computationally because the Steiner tree problem is known to be NP-hard, and is known to be computationally intractable even for relatively small sets with, say, 15 points. On the other hand, it was shown by Rayward-Smith [29, 30], Waxman and Imase [35], and Winter and Smith [38] that the quantity  $F_1^c$  defined above provides a basis for an accurate polynomial approximation algorithm for a variant of the Steiner tree problem. Due to this fact and since the quantities  $F_\alpha^c$  can be computed efficiently, we use them here as the basis of the adjusted averaging method.

#### 3.2 Adjusted Weighting Method (AWM)

This technique is a natural extension of the  $k$ -nearest neighbor classification method. The distance of a query point to a class is computed as the weighted average of the distances to the points of that class in the training set. As in the nearest neighbors method, we assume that the classification decision is determined by close neighbors of the query point. The distance of each query point to a point in a class is weighted by a factor which is geometrically decreasing as the class points get farther apart.

More formally, let

$$F_{\gamma}^c(q) = \sum_{i=1}^{\gamma} w^i d(q, p_i^c)$$

for an appropriately chosen integer  $\gamma$ ,  $1 \leq \gamma \leq n^*$ , and for some appropriately chosen weighting factor  $w \leq 1$ . We have found that  $w = 0.75$  is a good choice. A query point  $q$  is considered to belong to  $S_c$  if  $F_{\gamma}^c(q) \leq F_{\gamma}^{c'}(q)$  for all  $c'$ ,  $1 \leq c' \leq m$ .

### 3.3 Truncated Potentials Method (TPM)

This technique is based on the idea developed in the *method of potential functions* in pattern recognition (see [2] and [3]), combined with the approach used in the nearest neighbors algorithm. The points of the training set are viewed in this approach as "centers of attraction", whose attraction is quantified by a certain smooth potential function. The total attraction of a query point to a class is computed as the sum of attractions to the points of that class in the training set. The query point is then classified as belonging to the class to which it has the strongest attraction.

A potential function should be rapidly decreasing with the increase of the distance between the query point and the points in the training set (centers of attraction). It is natural to use exponential functions to construct such potentials.

It is assumed, as in the nearest neighbors method, that the classification decision is determined by close neighbors of the query point. Therefore, the total potential (attraction) for each class can be assumed to include only a fixed number of terms, corresponding to the potentials produced by the closest neighbors of the query point in the class. The potential "truncated" in such a way prevents the classification outcome from being excessively influenced by unbalance in the number of points in the different classes in the training set.

Formally, let

$$F_{\beta}^c(q) = \sum_{i=1}^{\beta} e^{-d(q, p_i^c)} \quad (1)$$

for an appropriately chosen integer  $\beta$ ,  $1 \leq \beta \leq n^*$ . A query point  $q$  is considered to belong to  $S_c$  if  $F_{\beta}^c(q) \geq F_{\beta}^{c'}(q)$  for all  $c'$ ,  $1 \leq c' \leq m$ .

### 3.4 Convex Containment Method (CCM)<sup>2</sup>

In most applications it is natural to expect that a point contained between two points from the same class will also belong to that class. A complicating factor, of course, is that the same query point may be contained between various pairs of points, and these pairs may belong to different classes. It should also be taken into account that usually the influence of a pair of points "closely" containing the query point is stronger than the influence of a pair of points farther away from the query point.

In order to formalize the discussion above, we need a definition of "betweenness". A straightforward definition of a point  $q$  being "between" the points  $p_i$  and  $p_j$  in a Euclidean space is given by

$$d(p_i, p_j) = d(p_i, q) + d(q, p_j).$$

To avoid the rigidity of this (rarely applicable) condition, we shall define betweenness here by using the following relaxation of the above:

$$wd(p_i, p_j) \geq d(p_i, q) + d(q, p_j), \quad (2)$$

<sup>2</sup>For a different use of convexity for classification see [4].

where  $w \geq 1$  is an appropriately chosen weighting factor. Obviously, the points in a plane satisfying the inequality (2) fill an ellipse.

Let  $\Pi^c(q)$  denote the set of pairs of points  $p_i^c, p_j^c$  from the class  $S_c$  that satisfy the last condition. In order to emphasize the role of containing pairs in the proximity of the query point, we shall classify a query point  $q$  as belonging to  $S_c$  if  $F_\pi^c(q) \leq F_\pi^{c'}(q)$  for all  $c', 1 \leq c' \leq m$ , where

$$F_\pi^c(q) = \sum_{(p_i^c, p_j^c) \in \Pi^c(q)} 2^{-d(p_i^c, p_j^c)}. \quad (3)$$

Clearly, in the formula (3) the choice of the base 2 instead of  $e$  (as in (1)) is equivalent to multiplying all distances by a constant close to 1, and therefore has no significant effect on the results.

#### 4. DATA SETS

All our real-life data sets are available from the *Machine Learning Repository*<sup>3</sup> of the Computer Science Department of the University of California at Irvine [20]. This choice was mainly based on the fact that this repository is well-known in the machine learning community and it has been widely used to compare various learning and classification algorithms. Moreover, these datasets span a wide variety of different application areas and most of them are sufficiently large to allow meaningful conclusions.

##### 4.1 Breast Cancer Diagnosis (Wisconsin)

This database was submitted to the Irvine Repository by Mangasarian and Bennett. The database contains 699 cytological tests described by 9 integer attributes with values between 1 and 10. 16 of these tests are incomplete. The outcome is a binary variable indicating the benign or malignant nature of the tumor.

##### 4.2 Classifying Irises

This is perhaps the best known database to be found in the pattern recognition literature. Fisher's paper [14] is a classic in the field and is referenced frequently to this day. The data set contains 3 classes of 50 instances each. Instances are described by 4 numerical attributes. Each class refers to a type of iris plant. One class is linearly separable from the other 2; the latter are not linearly separable from each other. This is an exceedingly simple domain.

Weiss and Kapouleas [36] obtained accuracies of 96.7% and 96.0% on this data with back propagation and 1-NN respectively.

##### 4.3 Housing Costs in Boston

This database was submitted to the Irvine repository by Harrison and Rubinfeld [18]. It contains 506 entries describing properties of houses in the suburbs of Boston using one binary and 12 continuous attributes. Houses are classified into two groups depending on whether their price exceeds the threshold value of \$ 21,000.

##### 4.4 Diabetes Diagnosis (Pima Indians)

This database originally owned by the National Institute of Diabetes and Digestive and Kidney Diseases was set up to investigate whether the patients show signs of diabetes according to the World Health Organization criteria. The population examined consisted of female Pima Indians, aged 21 or over, living near Phoenix, Arizona, USA. It contains 768 instances, each described by 8 continuous variables and a binary classification.

<sup>3</sup><ftp://ftp.ics.uci.edu/pub/machine-learning-databases>

#### 4.5 Credit Cards (Australian)

This database concerns credit card applications and was submitted to the Irvine repository by Quinlan [27, 28]. All attribute names and values have been changed to meaningless symbols to protect confidentiality of the data.

This database is interesting because there is a good mix of attributes – continuous, nominal with small numbers of values, and nominal with larger numbers of values. There are also a few missing values (37 cases), which have been removed. Our experiments involved the remaining 653 instances with 15 attributes each.

Carter and Catlett [8] reported an 85.5% correct prediction rate, when using 71% of all 690 instances as the training set.

#### 4.6 German Credit (Statlog)

This data set contains data used to evaluate credit applications in Germany. It has 1000 instances. We used a version of this data set that was produced by Strathclyde University. In this version each case is described by 24 continuous attributes. There are no missing values.

#### 4.7 Labor Negotiations

This data set includes all collective agreements reached in the business and personal services sector for locals with at least 500 members (teachers, nurses, university staff, police, etc) in Canada in 1987 and the first quarter of 1988. It contains 57 instances, each described by 16 attributes. There are no missing values. The data has a binary classification depending on whether or not a contract was considered acceptable.

#### 4.8 Soybean

The soybean database is the original data for soybean disease diagnosis [21]. It contains 630 instances of 15 disease classes. Each instance is described using 35 attributes. About 5% of attribute values are missing.

#### 4.9 Voting

This database includes votes for each of the 1984 U.S. House of Representatives Congressmen on 16 key votes [32].

The database contains 435 instances corresponding to 267 Democratic and 168 Republican Congressmen. Each instance involves 16 binary variables and 2 classes (party affiliation: democrat or republican). About 6% of attribute values are missing. One of the attributes (*physician-fee-freeze*) provides a clear-cut classification. In several applications, this attribute was eliminated in order to make the problem more interesting (or less trivial) [7, 22].

### 5. EXPERIMENTAL METHOD

In order to evaluate the distance-based methods described in Sections 2 and 3, we followed the cross-validation approach. In its simplest form, this approach consists of partitioning a database of classified points into two subsets: the *training subset*  $S$  and the *control subset*  $Q$  of *query points*. Each query point from the control subset  $Q$  is classified by the method under evaluation. This classification is then compared with the classification indicated in the data base. The ratio of the number of correctly classified query points to the total number of query points is taken as the estimate of the accuracy of the method under evaluation.

To reduce the variance of this estimate, the cross-validation procedure is usually repeated several times, with different subsets of points acting as training and control subsets. This can be achieved for example by using *k-fold cross-validation*, which consists of the following steps.



Data Set	AAM	AWM	TPM
Cancer	96.9 $\pm$ 0.2	97.2 $\pm$ 0.2	97.1 $\pm$ 0.3
Iris	95.4 $\pm$ 0.7	94.1 $\pm$ 0.4	95.3 $\pm$ 0.6
Housing	84.3 $\pm$ 0.8	84.9 $\pm$ 0.7	85.3 $\pm$ 1.0
Diabetes	75.3 $\pm$ 0.7	74.9 $\pm$ 0.5	75.5 $\pm$ 0.6
Credit Card (A)	87.5 $\pm$ 0.4	86.9 $\pm$ 0.6	87.3 $\pm$ 0.4
German Credit	73.7 $\pm$ 0.6	73.8 $\pm$ 0.7	73.4 $\pm$ 0.6
Labor	88.1 $\pm$ 3.1	88.8 $\pm$ 2.7	90.2 $\pm$ 2.4
Soybean	90.0 $\pm$ 0.5	89.8 $\pm$ 0.6	90.7 $\pm$ 0.6
Voting/All	91.9 $\pm$ 0.6	92.8 $\pm$ 0.6	92.6 $\pm$ 0.6
Voting/Red	89.2 $\pm$ 0.5	89.6 $\pm$ 0.7	89.9 $\pm$ 0.7

Data Set	CCM	k-NN	Parzen	Best Known
Cancer	97.2 $\pm$ 0.2	97.4 $\pm$ 0.2	97.0 $\pm$ 0.2	96.2 [23]
Iris	95.4 $\pm$ 0.8	95.5 $\pm$ 0.6	94.6 $\pm$ 0.7	98.0 [36]
Housing	86.5 $\pm$ 0.6	84.3 $\pm$ 1.0	83.4 $\pm$ 0.8	83.2 [23]
Diabetes	74.8 $\pm$ 0.7	75.4 $\pm$ 0.7	74.1 $\pm$ 0.5	76.0 [33] 78.0 [6]
Credit Card (A)	87.1 $\pm$ 0.4	87.7 $\pm$ 0.4	87.2 $\pm$ 0.4	85.5 [8] 87.0 [6]
German Credit	73.7 $\pm$ 0.7	73.4 $\pm$ 0.6	73.3 $\pm$ 0.7	
Labor	89.8 $\pm$ 3.2	85.4 $\pm$ 2.7	74.8 $\pm$ 3.9	90.0 [5]
Soybean	89.3 $\pm$ 0.6	89.9 $\pm$ 0.7	89.7 $\pm$ 0.6	87.0 [22]
Voting/All	95.8 $\pm$ 0.5	92.8 $\pm$ 0.5	91.8 $\pm$ 0.6	95.6 [19]
Voting/Red	89.5 $\pm$ 0.4	89.6 $\pm$ 0.6	88.9 $\pm$ 0.5	89.4 [19]

**Table 1:** Distance-based classifiers - manual parameter selection

- Randomly partition the data base into  $k$  equal-size disjoint subsets  $Q_1, Q_2, \dots, Q_k$ .
- Use  $S_i = S \setminus Q_i$  as a training subset and  $Q_i, i = 1, 2, \dots, k$  as a control subset.
- Sum up the number of correctly classified points (each point of the entire data base acts as a query point exactly once). The classification accuracy is obtained by dividing this sum by the size of the data base.

Furthermore,  $k$ -fold cross-validation can be repeated several times, each time with a different random partition of the data set. The average accuracy is then reported as the overall accuracy.

All distance-based methods are parametric in the sense that their performance depends on some integer- or real-valued parameter. In our initial evaluation of each method, its parameter value was chosen "manually". More specifically, for each parameter value chosen from an appropriately broad range, five 5-fold cross-validations (i.e., 25 runs, each with 80% of points acting as the training subset, and the remaining 20% acting as the control subset) were carried out. Based on these results, the "best" parameter value was fixed and thirty 5-fold cross-validations (150 runs) were carried out to determine average accuracy.

In later stages of our experiments, we used a more sound "automatic" way of selecting parameter values. More specifically, five 5-fold cross-validations determining the "best" parameter value were omitted. Instead, within each of five runs of the thirty 5-fold cross-validations, the training subset  $S_i$  was subjected to one 4-fold cross-validation (4 runs) for each parameter value chosen from an appropriately broad range. The best

Data Set	AAM	AWM	TPM
Cancer	$96.7 \pm 0.2$	$96.9 \pm 0.3$	$96.8 \pm 0.3$
Iris	$92.2 \pm 2.1$	$94.1 \pm 0.4$	$95.3 \pm 0.6$
Housing	$83.6 \pm 1.1$	$84.2 \pm 1.0$	$84.0 \pm 1.0$
Diabetes	$74.8 \pm 0.6$	$74.6 \pm 0.7$	$74.9 \pm 0.7$
Credit Card (A)	$86.9 \pm 0.5$	$86.5 \pm 0.8$	$86.6 \pm 0.8$
German Credit	$73.5 \pm 0.8$	$73.7 \pm 0.8$	$73.7 \pm 0.8$
Labor	$86.7 \pm 3.7$	$87.9 \pm 2.7$	$89.5 \pm 2.3$
Soybean	$83.4 \pm 2.1$	$89.6 \pm 0.6$	$90.5 \pm 0.4$
Voting/All	$91.6 \pm 0.4$	$92.6 \pm 0.8$	$92.5 \pm 0.6$
Voting/Red	$89.0 \pm 0.7$	$89.2 \pm 0.7$	$88.3 \pm 0.8$

Data Set	CCM	k-NN	Parzen	Best Known
Cancer	$97.0 \pm 0.2$	$97.3 \pm 0.3$	$96.4 \pm 0.3$	96.2 [23]
Iris	$95.0 \pm 1.0$	$94.7 \pm 0.8$	$91.8 \pm 1.0$	98.0 [36]
Housing	$84.4 \pm 0.6$	$83.4 \pm 1.1$	$83.1 \pm 1.1$	83.2 [23]
Diabetes	$74.1 \pm 0.6$	$74.6 \pm 0.8$	$73.4 \pm 0.9$	76.0 [33] 78.0 [6]
Credit Card (A)	$86.4 \pm 0.6$	$86.5 \pm 0.6$	$86.5 \pm 0.6$	85.5 [8] 87.0 [6]
German Credit	$73.4 \pm 0.8$	$73.1 \pm 0.7$	$71.7 \pm 0.7$	
Labor	$88.4 \pm 4.1$	$83.5 \pm 4.6$	$69.9 \pm 5.6$	90.0 [5]
Soybean	$88.9 \pm 0.5$	$89.3 \pm 0.6$	$86.7 \pm 0.6$	87.0 [22]
Voting/All	$94.1 \pm 0.6$	$92.2 \pm 0.6$	$89.3 \pm 0.9$	95.6 [19]
Voting/Red	$89.1 \pm 0.7$	$88.9 \pm 1.0$	$88.2 \pm 0.7$	89.4 [19]

**Table 2:** Distance-based classifiers – automatic parameter selection

parameter value (the one with the highest average accuracy over 4 runs) was then used to classify instances in the control subset  $Q_i$ .

We used the  $L_1$ -metric to measure distances between query points and points in the training set. Replacement of the  $L_1$ -metric by the  $L_2$ -metric did not seem to improve the performance on the tested data sets for larger data bases. However, for smaller data bases (such as Labor Negotiations) we observed considerably lower accuracy (up to 10%) for all distance methods when using the  $L_2$  metric.

No attempt to preprocess data sets was made. Numerical attributes were simply normalized between 0 and 1. Normalization based on frequencies did not result in any significant improvement for the data sets we tested.

Nominal attributes were arbitrarily ordered, replaced by consecutive integers 0, 1, ..., and normalized. Changes in the ordering did not improve the performance of our method on the data sets we tested. The replacement of every nominal attribute  $x$  taking the values  $v_1, v_2, \dots, v_m$  by  $m$  0-1 attributes  $b_i^x$  (defined by  $b_i^x = 1$  iff  $x = v_i$ ,  $i = 1, \dots, m$ ) did not lead to significant improvements.

## 6. RESULTS

Table 1 shows how the methods perform when using the manual way of selecting parameter values. Each entry is an average over 150 runs (30 runs, each with 5-fold cross-validation). It should be noted that the values in Table 1 only provide upper bounds on the real accuracies of the methods since the parameters were selected in such

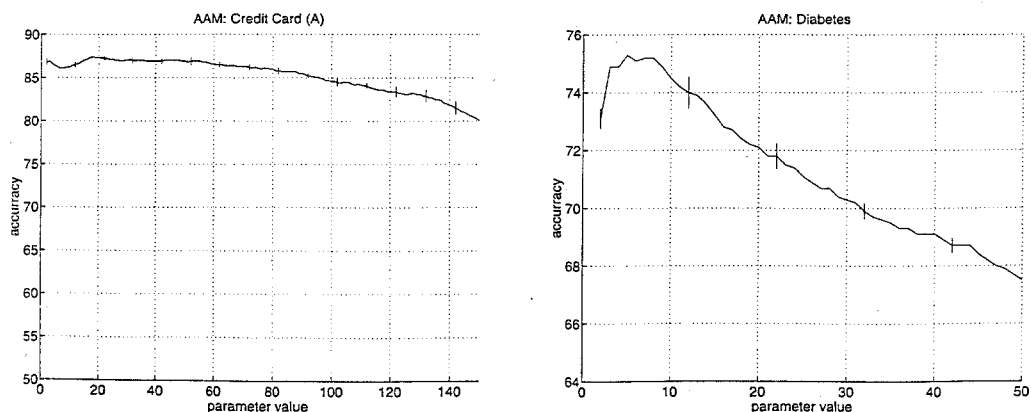


Figure 1: Robustness of AAM,  $1 \leq \alpha \leq x$

a was as to achieve the best possible results using the knowledge of classifications of the testing set. In a more realistic situation this knowledge not being available, the accuracies will be somewhat lower. The values shown in Table 2 are real estimates of accuracy.

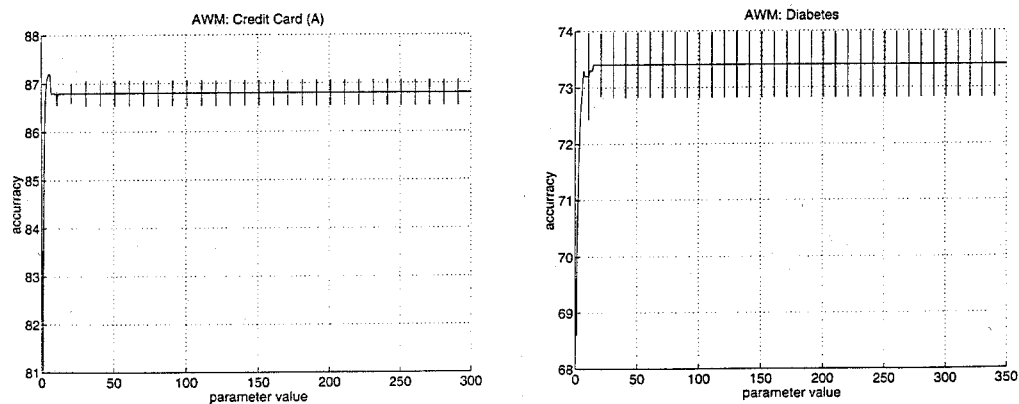
Table 2 shows how the methods perform when using the automatic way of selecting parameter values. Each entry is an average over 150 runs (30 5-fold cross-validation runs). As it could be expected, the accuracy drops slightly. It can be seen that for most data sets the accuracy of the new methods is comparable with the  $k$  nearest neighbors and the Parzen windows approaches. Some better results are reported in [20] without specification of any details of experimentation methods. If the accuracy results from [20] are disregarded (as no reference to the origin of those numbers is given there), the new methods are comparable with the best classification methods reported in the literature.

We also tested how the distance-based classification methods perform for various parameter values. For each method and for each parameter value five 5-fold cross-validations (25 runs) on the Australian credit card data base were used to obtain average accuracies. It can be seen from Figures 1 – 6 that AWM and TPM are very stable. AAM seems to remain stable for parameter values between 2 and 50. It should be noted that the computing time of all the methods increases with the increase of the parameter values.

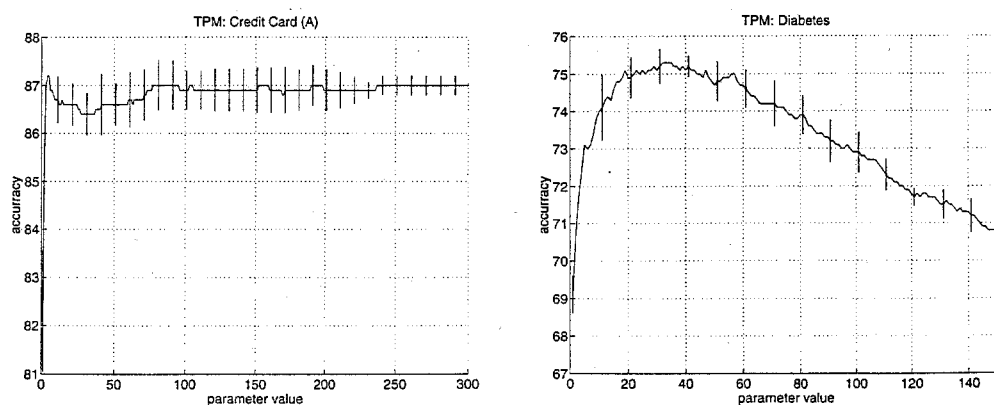
## 7. CONCLUSIONS

This paper introduces a number of simple classification methods based on the use of distance measures from the query points to the points in the training set. These methods are easy to implement partly because they require little or no preprocessing. At the same time, all these techniques are quite robust and very competitive in terms of their classification accuracy.

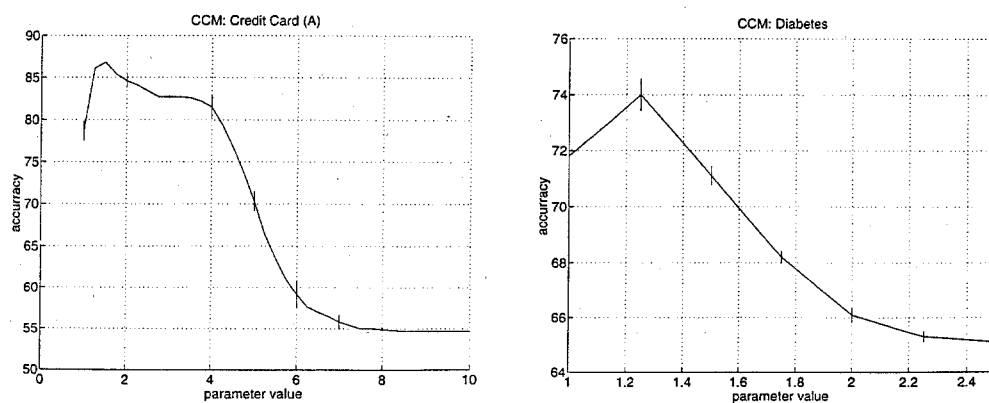
The techniques introduced in this paper include the adjusted averaging method (AAM), the adjusted weighting method (AWM), the truncated potentials method (TPM) and the convex containment method (CCM). A number of classification techniques of similar nature are well studied in the literature. Among the best known methods, we mention the  $k$ -nearest neighbors and the Parzen windows. Each method classifies a new query point using a special numerical measure of proximity of this query point to each class. The proximity measures in each method, computed using the distances from this query point to the points in the training set, depend on a single numerical parameter.



**Figure 2:** Robustness of AWM,  $1 \leq \gamma \leq x$



**Figure 3:** Robustness of TPM,  $1 \leq \beta \leq x$



**Figure 4:** Robustness of CCM,  $1 \leq w \leq x$

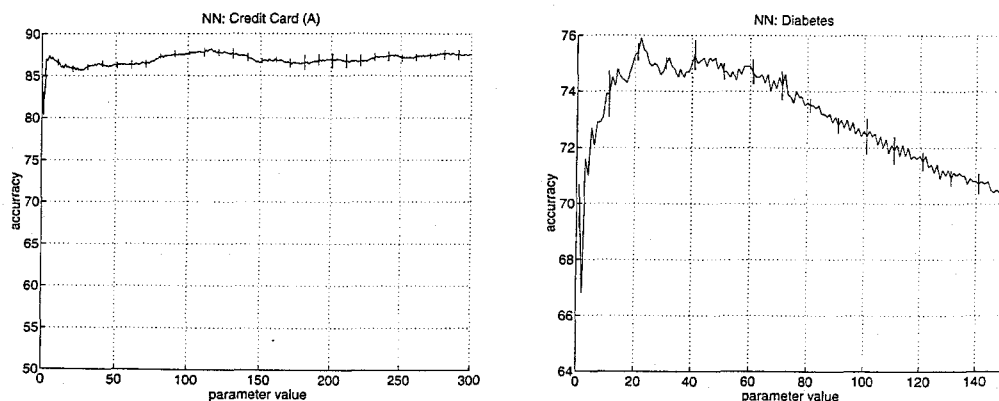


Figure 5: Robustness of NN,  $1 \leq k \leq x$

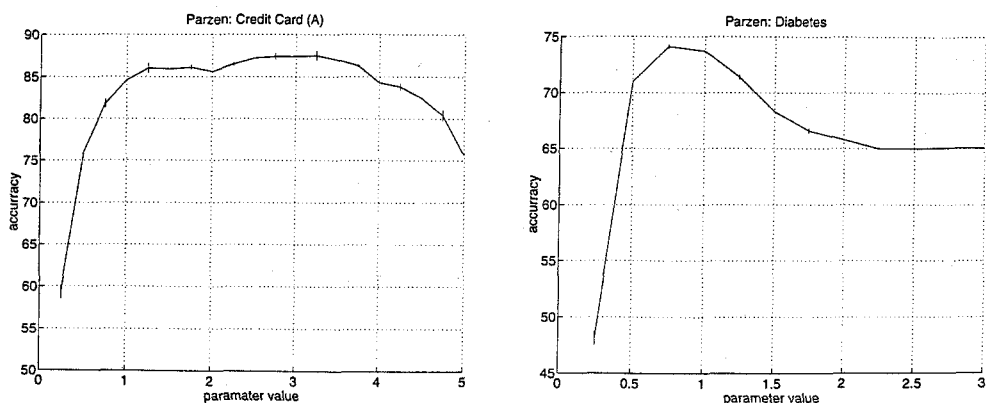


Figure 6: Robustness of Parzen,  $1 \leq r \leq x$

The value of this parameter is chosen automatically by each algorithm to achieve the best classification accuracy.

While all these techniques can be based on any distance measure between points, the  $L_2$  and  $L_1$  metrics are the most frequently used ones. Our experience seems to confirm the comparable nature of the  $L_2$  and  $L_1$  metrics in distance-based methods, with a slightly superior performance when the  $L_1$  metric is used.

While all the methods measure proximity using distances, the particular definitions of proximities differ substantially from one method to another. When computing proximity measures for a query point, each method—except for CCM—arranges the points in the training set in ascending order of distances to the query point, and uses an initial segment of this ordering in the actual computation of the proximity measure. The number of points of the training set in this initial segment is the numerical parameter used in AWM, TPM and k-NN methods. After the value of this parameter is chosen, all these three methods use a constant number of training set points in the initial segment for computing proximity measures for each query point. For the AAM and Parzen methods, the number of points in the initial segment is query point dependent. In CCM, the proximity measure is computed using the distances between those pairs of points of the same class in the training set which “contain” the query point. The distances from the

query point to the points in the training set are not used explicitly in the computation of this proximity measure.

A surprising conclusion of this study is the fact that in spite of the considerably different ways in which proximities are defined, the performances of the various procedures examined in this paper are very similar. It is natural to explain this striking uniformity by the common element in all these methods: the use of distances in classification.

### ACKNOWLEDGMENT

The partial support of ONR (Grants N00014-92-J1375 and N00014-92-J4083) and the Danish Research Council is gratefully acknowledged. The authors would like to thank the anonymous referees for useful suggestions that helped improve the presentation of this paper. This research was carried out during Pawel Winter's visit to DIMACS and RUTCOR whose hospitality and support are gratefully appreciated.

### Appendix

#### Lemma 1.

The sequence

$$f_{\alpha}^c(\alpha + 1, q), f_{\alpha}^c(\alpha + 2, q), \dots, f_{\alpha}^c(n_c, q)$$

is unimodal for any  $\alpha \geq 0$ .

**Proof.**

We shall show that

$$f_{\alpha}^c(\alpha + k, q) \leq f_{\alpha}^c(\alpha + k + 1, q) \implies f_{\alpha}^c(\alpha + k + 1, q) \leq f_{\alpha}^c(\alpha + k + 2, q)$$

for any integer  $k \geq 0$ . This follows almost directly from the following straightforward algebraic manipulation:

$$\begin{aligned} f_{\alpha}^c(\alpha + k + 2, q) - f_{\alpha}^c(\alpha + k + 1, q) &= \frac{\sum_{j=1}^{\alpha+k+2} d(q, p_j)}{k+2} - \frac{\sum_{j=1}^{\alpha+k+1} d(q, p_j)}{k+1} \\ &= \frac{\sum_{j=1}^{\alpha+k+1} d(q, p_j)}{k+2} - \frac{\sum_{j=1}^{\alpha+k+1} d(q, p_j)}{k+1} + \frac{d(q, p_{\alpha+k+2})}{k+2} \\ &= \frac{(k+1) \sum_{j=1}^{\alpha+k+1} d(q, p_j) - (k+2) \sum_{j=1}^{\alpha+k+1} d(q, p_j) + (k+1)d(q, p_{\alpha+k+2})}{(k+1)(k+2)} \\ &= \frac{(k+1)d(q, p_{\alpha+k+2}) - \sum_{j=1}^{\alpha+k+1} d(q, p_j)}{(k+1)(k+2)} \\ &= \frac{(k+1)d(q, p_{\alpha+k+2}) - \sum_{j=1}^{\alpha} d(q, p_j) - \sum_{j=\alpha+1}^{\alpha+k+1} d(q, p_j)}{(k+1)(k+2)}. \end{aligned}$$

This term is non-negative iff

$$(k+1)d(q, p_{\alpha+k+2}) - \sum_{j=1}^{\alpha} d(q, p_j) - \sum_{j=\alpha+1}^{\alpha+k+1} d(q, p_j) \geq 0$$

By assumption,  $f_{\alpha}^c(\alpha + k + 1, q) - f_{\alpha}^c(\alpha + k, q) \geq 0$ , i.e.

$$kd(q, p_{\alpha+k+1}) - \sum_{j=1}^{\alpha} d(q, p_j) - \sum_{j=\alpha+1}^{\alpha+k} d(q, p_j) \geq 0,$$

which is equivalent to

$$(k+1)d(q, p_{\alpha+k+1}) - \sum_{j=1}^{\alpha} d(q, p_j) - \sum_{j=\alpha+1}^{\alpha+k+1} d(q, p_j) \geq 0.$$

Since  $d(q, p_{\alpha+k+1}) \leq d(q, p_{\alpha+k+2})$ , we obtain

$$(k+1)d(q, p_{\alpha+k+2}) - \sum_{j=1}^{\alpha} d(q, p_j) - \sum_{j=\alpha+1}^{\alpha+k+1} d(q, p_j) \geq 0,$$

as desired. •

## BIBLIOGRAPHY

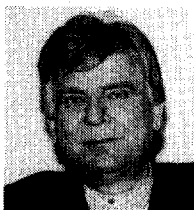
- [1] D.W. Aha, D. Kibler and M.K. Albert, Instance-based learning algorithms, *Machine Learning*, **6** (1991), 37-66.
- [2] M.A. Aiserman, E.M. Braverman and L.I. Rosonoer, *The Method of Potential Functions in the Theory of Machine Learning* (in Russian), Nauka, Moscow, USSR (1970).
- [3] O.A. Bashkirov, E.M. Braverman and I.B. Muchnik, Potential function algorithms for pattern recognition learning machines (in Russian), *Avtomatika i Telemekhanika*, **25** (1964) 5, 692-695.
- [4] P. Belfays and J.-P. Rasson, A new geometric discriminant rule, *Computational Statistics Quarterly*, **2** (1) (1985), 15-30.
- [5] F. Bergandano, S. Matwin, R.S. Michalski and J. Zhang, Learning two-tiered descriptions of flexible concepts: The Poseidon system, *Machine Learning*, **8** (1992), 5-44.
- [6] P. Brazdil and J. Gama, Evaluation/Characterization of Classification Algorithms, LIACC, University of Porto, Rua Campo Alegre 823 4150 Porto, Portugal [<http://www.up.pt/liacc/ML/statlog/index.html>].
- [7] W. Buntine and T. Niblett, A further comparison of splitting rules for decision-tree induction, *Machine Learning*, **8** (1992), 75-86.
- [8] C. Carter and J. Catlett, Assessing credit card applications using machine learning, *IEEE Expert*, Fall 1987, 71-79.
- [9] S. Cost and S. Salzberg, A weighted nearest neighbor algorithm for learning with symbolic features, *Machine Learning*, **10** (1993), 57-78.
- [10] T.M. Cover and P.E. Hart, Nearest neighbor pattern classification, *IEEE Trans. Information Theory*, **IT-13** (1967), 21-27.
- [11] B. V. Dasarathy, *Nearest Neighbor (NN) Norms: NN Pattern Classification Techniques*, (1991), Los Alamitos, CA: IEEE Computer Society Press.
- [12] P.A. Devijver and J. Kittler, *Pattern Recognition: A Statistical Approach*, Prentice Hall, Englewood Cliffs, New Jersey (1982).
- [13] S. Dudani, The distance-weighted k-nearest-neighbor rule, *IEEE Transactions on Systems, Man, and Cybernetics*, **4**, 325-327.
- [14] R. A. Fisher, The use of multiple measurements in taxonomic problems, *Annual Eugenics*, **7** (1936), 179-188, also in *Contributions to Mathematical Statistics*, John Wiley, NY (1950).
- [15] F.P. Fisher and E.A. Patrick, A preprocessing algorithm for nearest neighbor decision rules, *Proceedings of the National Electronic Conference*, **26** (1970), 481-485.
- [16] E.Fix and J.L.Hodges, *Discriminatory Analysis - Nonparametric Discrimination: Consistency Properties*, USAF School of Aviation Medicine, Randolph Field, Tex., Project 21-49-004, Report No. 4 (1951), 261-279.
- [17] K. Fukunaga and T.E. Flick, An optimal global nearest neighbor metric, *IEEE Trans. PAMI*, **6** (1984), 314-318.
- [18] D. Harrison and D.L. Rubinfeld, Hedonic prices and the demand for clean air, *J. Environ. Economics and Management*, **5** (1978), 81-102.
- [19] R.C. Holte, Very simple classification rules perform well on most commonly used datasets, *Machine Learning*, **11** (1993), 63-91.

- [20] C.J. Merz and P.M. Murphy, UCI Repository of Machine Learning Databases [<http://www.ics.uci.edu/mllearn/MLRepository.html>], Irvine, CA, University of California, Department of Information and Computer Science (1996).
- [21] R.S. Michalski and R.L. Chilausky, Knowledge acquisition by encoding expert rules versus computer induction from examples: A case study involving soybean pathology *Int. J. Man-Machine Studies*, **12** (1980), 63-87.
- [22] R.J. Mooney, Encouraging experimental results on learning CNF, *Machine Learning*, **19** (1995), 79-92.
- [23] S.K. Murthy, S. Kasif and S. Salzberg, A system for induction of oblique decision trees, *J. of AI Research*, **2** (1994), 1-32.
- [24] M.O. Noordewier, G.G. Towell and J.W. Shavlik, Training knowledge-based neural networks to recognize genes in DNA sequences, *Adv. in Neural Information Processing Systems*, **3** (1990).
- [25] E. Parzen, On estimation of a probability density function and mode, *Ann. Math. Stat.*, **33** (1962), 1065-1076.
- [26] B. Porter, R. Bareiss and R. Holte, Concept learning and heuristic classification in weak-theory domains, *Artificial Intelligence*, **45** (1990), 229-263.
- [27] J. R. Quinlan, Simplifying decision trees, *Int. J. Man-Machine Studies*, **27** (1987), 221-234.
- [28] J. R. Quinlan, *C4.5: Programs for Machine Learning*, Morgan Kaufmann, San Mateo, CA (1993).
- [29] V.J. Rayward-Smith, The computation of nearly minimal Steiner trees in graphs, *Int. J. Math. Educ. Sci. Technol.*, **14** (1983), 15-23.
- [30] V.J. Rayward-Smith and A. Clare, On finding Steiner vertices, *Networks*, **16** (1986), 283-294.
- [31] S. Salzberg, A nearest hyperrectangle learning method, *Machine Learning*, **6** (1991), 251-276.
- [32] J.C. Schlimmer, *Concept Acquisition Through Representational Adjustment*, Ph.D. Thesis, Dept. of Information and Computer Science, Univ. of California, Irvine, CA (1984).
- [33] J. Smith, J. Everhart, W. Dickson, W. Knowler and R. Johannes, Using the ADAP learning algorithm to forecast the onset of diabetes mellitus, *Proc. of the Symp. on Computer Applications and Medical Care*, (1988), 261-265.
- [34] G.G. Towell, J.W. Shavlik and M.O. Noordewier, Refinement of approximate domain theories by knowledge-based artificial neural networks, *Proc. of the 8-th Nat. Conf. on Art. Int., Boston, MA*, (1990), 861-866.
- [35] B.M. Waxman and M. Imase, Worst-case performance of Rayward-Smith's Steiner tree heuristics, *Inf. Process. Lett.*, **29** (1988), 283-287.
- [36] S. Weiss and I. Kapouelas, An empirical comparison of pattern recognition, neural nets, and machine learning classification methods, *Proc. of the 11-th IJCAI*, (1989), 781-787.
- [37] D. Wettschereck and T. Dietterich, An experimental comparison of the nearest-neighbor and nearest-hyperrectangle algorithms, *Machine Learning*, (1995), **19**, 5-27.
- [38] P. Winter and J. MacGregor Smith, Path-distance heuristics for the Steiner tree problem in undirected networks, *Algorithmica*, **7** (1992), 309-327.



**Oya Ekin** received B.S. degree from Cornell University, Ithaca, NY in Computer Science in 1988, M.S. degree from Bilkent University, Ankara, Turkey in Industrial Engineering in 1991 and Ph.D. degree from Rutgers University, NJ in Operations Research in 1997. She worked as a project manager in AT&T at Middletown, New Jersey between July 1997 and December 1997. Since January 1998, she is an assistant professor in the Industrial Engineering department at Bilkent University, Ankara, Turkey. Her current research interests lie in the field of combinatorial optimization with specialty in Boolean functions.





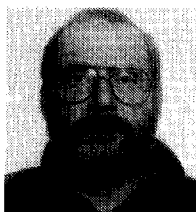
**Peter Hammer** received his doctorate in mathematics at the University of Bucharest, Romania, and honorary degrees from the Swiss Federal Institute of Technology (1986) and La Sapienza University of Rome, Italy (1998). Currently, he is the Director of RUTCOR, Rutgers University's Center for Operations Research, and is the Founder and Editor-in-Chief of the journals *Discrete Mathematics* and *Discrete Applied Mathematics* (Elsevier), the publication *Annals of Operations Research* (Baltzer Scientific), and the new SIAM series of *Monographs on Discrete Mathematics and Applications*. He has done extensive research on the theory and ap-

plications of Boolean and pseudo-Boolean functions in operations research and related areas. He has published seven books and is the author of over 170 papers. His interests center on discrete applied mathematics, and in particular on Boolean functions, logical analysis of data, and discrete optimization. Currently, his main interest centers on the use of Boolean techniques and combinatorial optimization in datamining. Dr. Hammer is a member of ACM, AMS, CORS, IEEE Computer Society, INFORMS, MAA, MPS, and SIAM.



**Alexander Kogan** received the M.S. degree in applied mathematics and operations research from the Moscow Institute of Physics and Technology (Phystech), and the Ph.D. degree in computer science from the USSR Academy of Sciences, Moscow, in 1984 and 1988, respectively. He is currently an Associate Professor of Accounting and Information Systems with the Faculty of Management, Rutgers University, Newark, NJ, and is also a member of RUTCOR - Rutgers University's Center for Operations Research, New Brunswick, NJ. His research interests are in the areas of expert systems, artificial intelligence, knowledge-based decision

support systems, accounting information systems, accounting problems of Internet infrastructure and electronic commerce, productivity accounting, logical analysis of data, Boolean functions, and reasoning under uncertainty. Dr. Kogan has published over forty technical papers in these areas. His research articles appear in scholarly journals (e.g. *Artificial Intelligence*, *IEEE Transactions on Knowledge and Data Engineering*, *Mathematical Programming*, *Decision Support Systems*, *Discrete Applied Mathematics*, *Information Processing Letters*, *Annals of Mathematics and Artificial Intelligence*, *International Journal of Accounting*, *Fuzziness and Knowledge-Based Systems*, *IS Audit & Control Journal*, *Management Accounting*, *Journal of Computational Mathematics and Mathematical Physics*, *Soviet Mathematics - Doklady*), conference proceedings, and topical volumes. Dr. Kogan is a member of AAA, AAAI, IEEE Computer Society, and INFORMS.



**Pawel Winter** is an Associate Professor in the Department of Computer Science at the University of Copenhagen in Denmark. He has earned a Master's Degree and Ph.D. from the University of Copenhagen. His current research interests include network design, computational geometry and computational biology.